

# Spezifikation von Anwendungskernen für große Informationssysteme

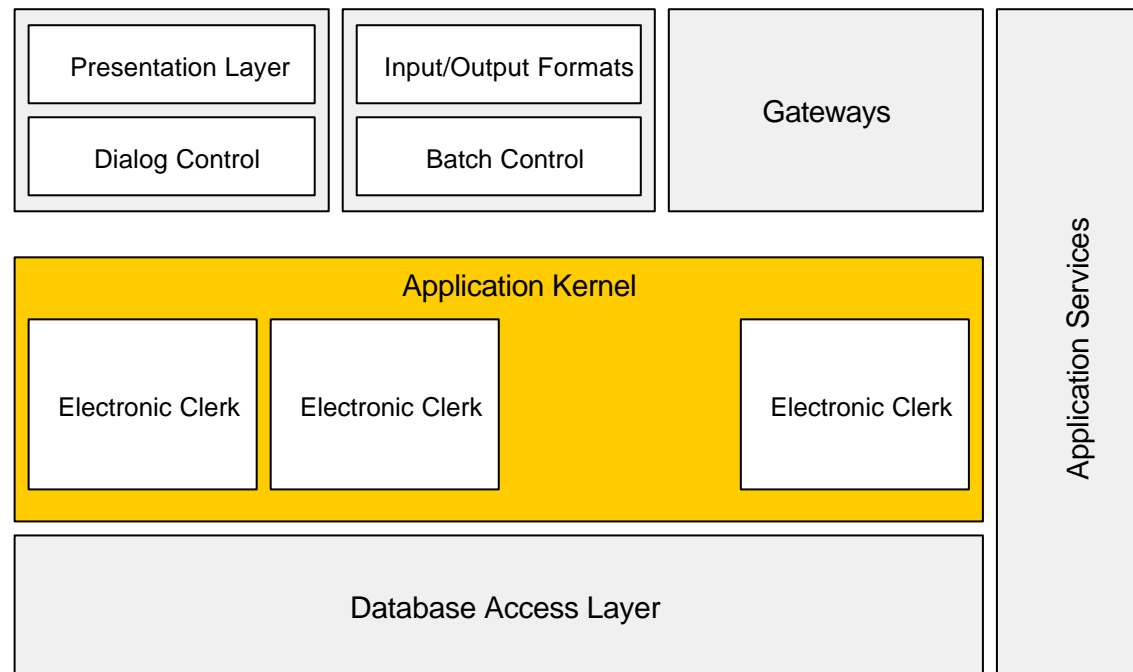
Vorlesung 4: Software-Engineering für große Informationssysteme  
TU-Wien, Sommersemester 2002  
Wolfgang Keller

# Überblick

- Anwendungskern – wo finde ich den?
- Spezifikation versus „Runterprogrammieren“ versus UML- Bilder
- Ein Beispiel für textuelle Spezifikation: Die Sachbearbeitermethode
- Die Sachbearbeitermethode und OO
  - Deltas zu UML und hartem OO
  - Die Sachbearbeitermethode und J2EE Patterns

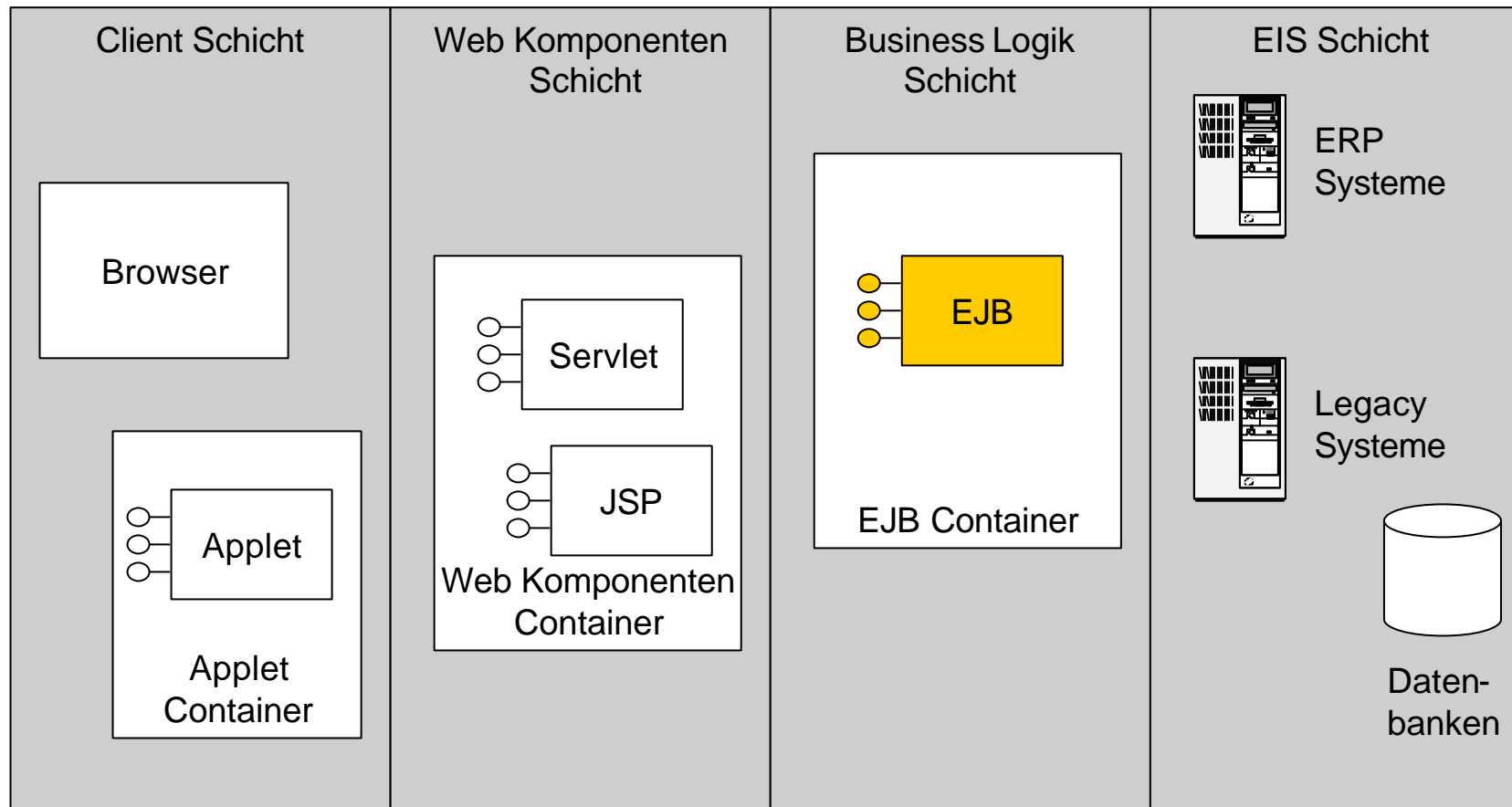
# Anwendungskern

## Wo finde ich den?



# Anwendungskern

## Wo finde ich den?



# Anwendungskern Verantwortlichkeiten

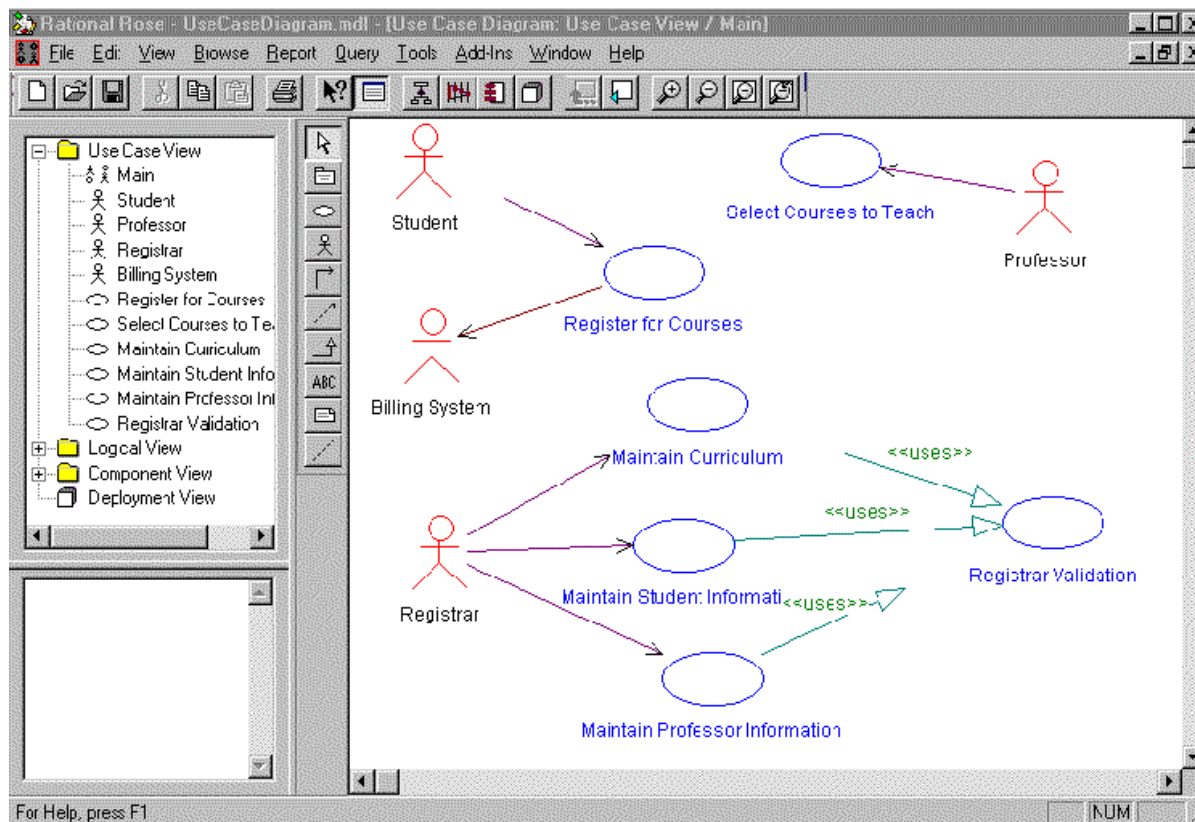


- Ein Anwendungskern implementiert fachliche Funktionalität
- Seine Abstraktionen sind „fachliche Objekte“ und er hat dementsprechend „fachliche Aufträge“.
- Der Anwendungskern muß auch ohne GUI oder Oberfläche funktionieren

# Spezifikation versus „Runterprogrammieren“ versus UML- Bilder

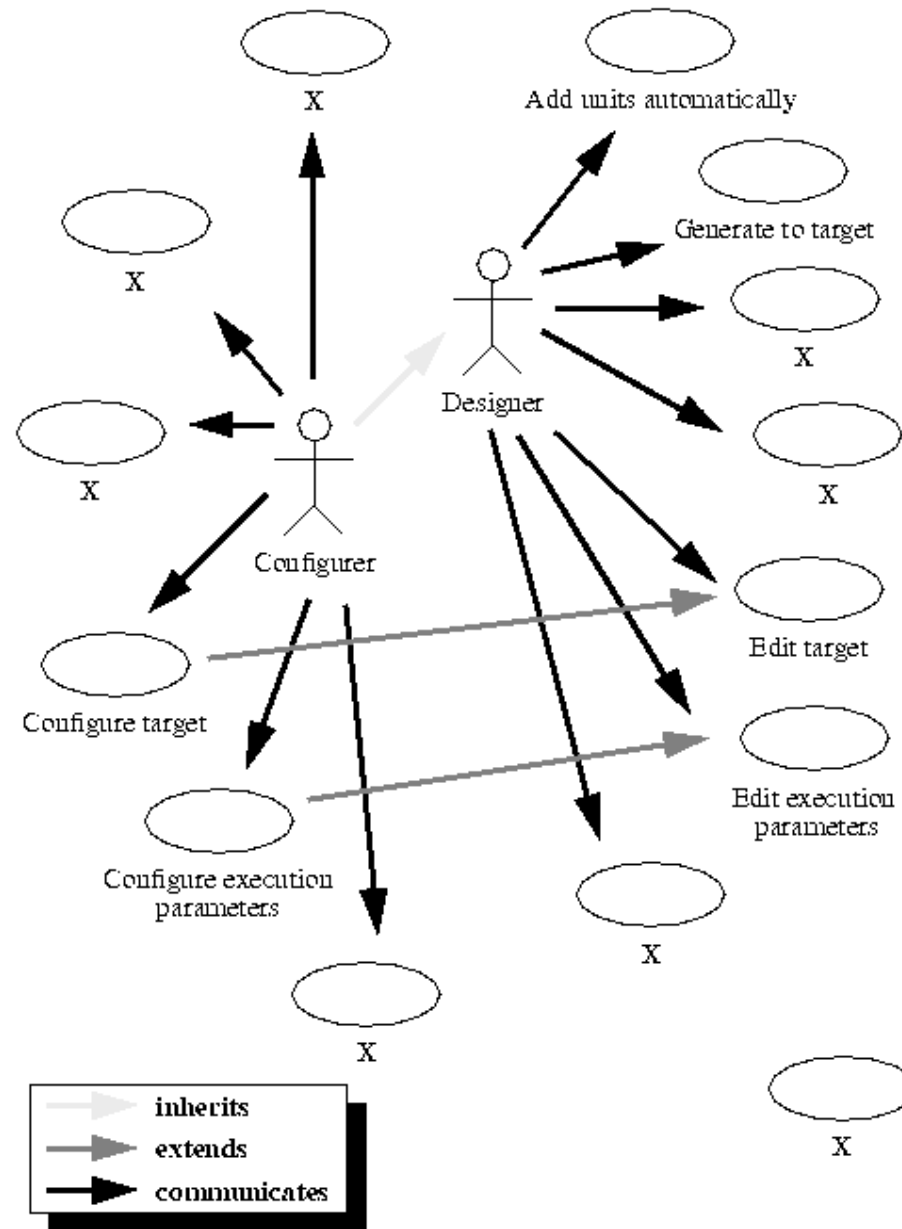
# Bilder ...

- ein Bild sagt mehr als 1000 Worte
- aber was sagen 1000 Bilder dieser Art ...



# Bilder ...

- noch ein schönes Beispiel ...
- Kapiieren Sie was???

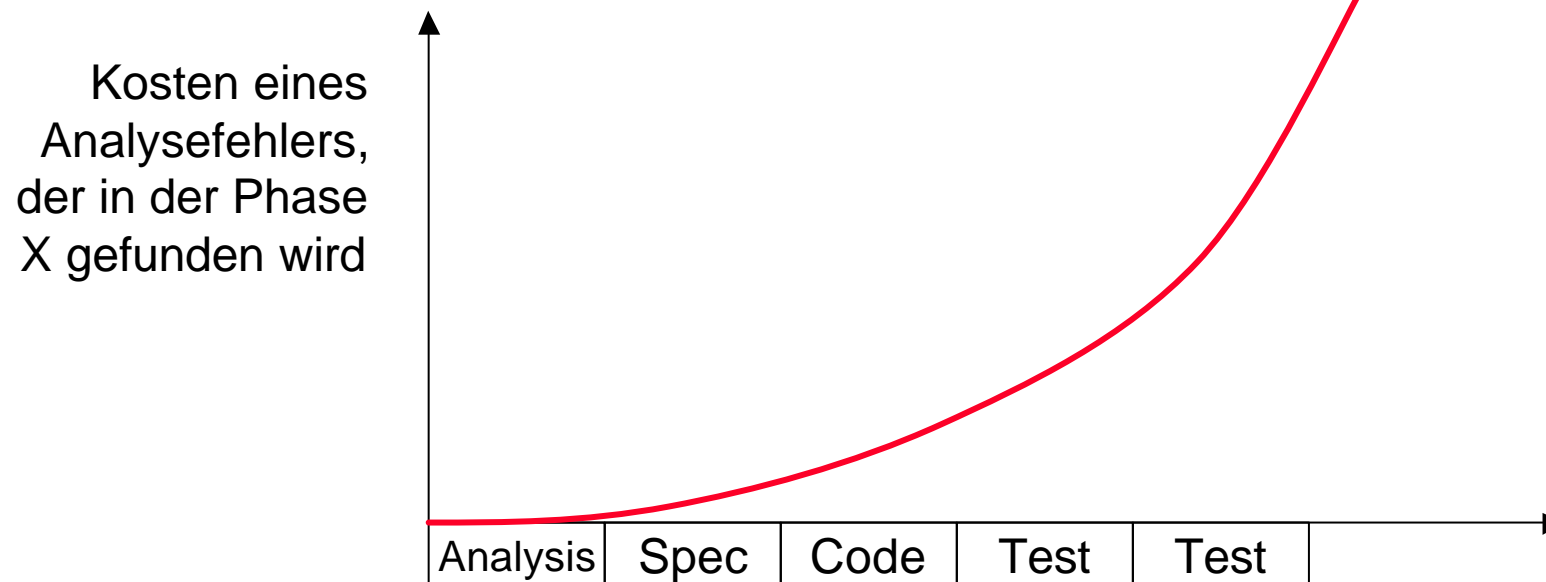


## einfach „Runterprogrammieren“?

- das geht solange man sich im Umfeld von XP bewegt
- Wir erinnern uns an die Voraussetzungen:
  - entsprechende Entwicklungsumgebung
  - Debuggen, Änderungen und Refactoring werden einfach unterstützt
  - Kurze Compile oder Change Zyklen
  - in großen Team nicht erprobt

## einfach „Runterprogrammieren“?

- wenn Sie die Voraussetzungen von XP oder agilen Prozessen nicht erfüllen, haben Sie es mit der Änderungskostenkurve zu tun



## Folgerung

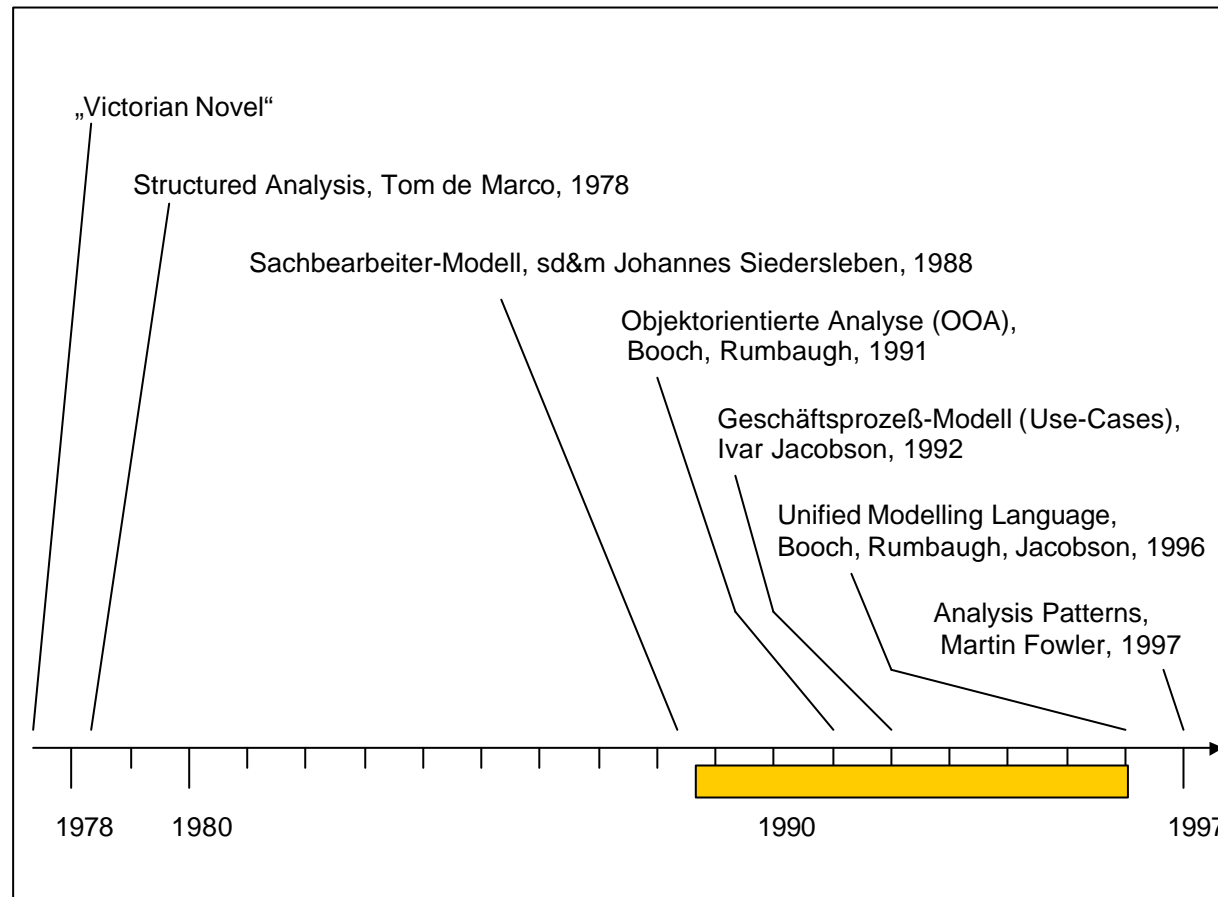
- einfach Runterprogrammieren: Für große Systeme wegen der Arbeitsteiligkeit nicht geeignet
  - und für „ältere Umgebungen“ mit langsamen Zyklen schon garnicht
- 1000 Bilder
  - es kann passieren, dass man dabei den Überblick verliert
- Also: Auf eine gesunde Mischung kommt es an aus
  - Bilder, Texten, Generierung ....
- Damit Sie die finden, heute mal ein Beispiel, wie man es rein textuell macht – einfach als Contrapunkt und weil das sonst selten gezeigt wird.

# Ein Beispiel für textuelle Spezifikation: Die Sachbearbeitermethode

# Spezifikation mit der Sachbearbeitermethode

- Wann war das „state of the art“?
- Was zeichnet eine gute Spezifikation aus

# Wo befinden wir uns dabei in der Geschichte der Software ..



Quelle sd&m

# Grundlage für Implementierung von Software



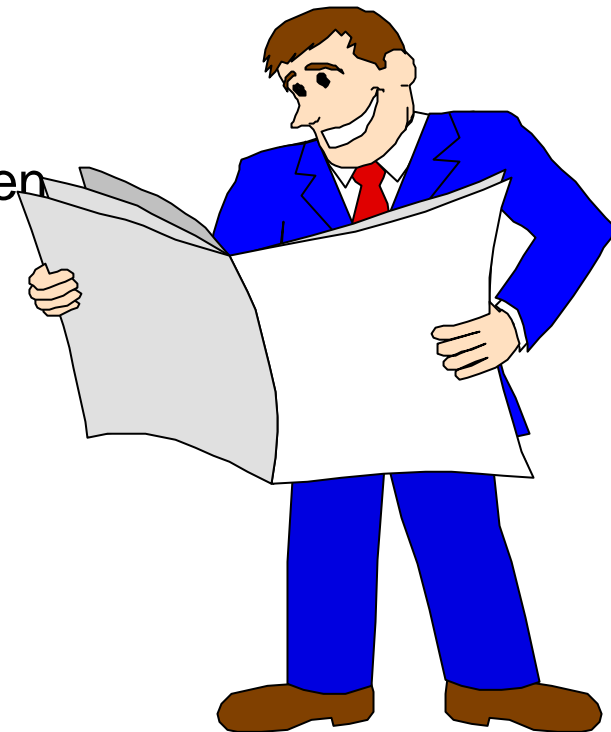
- Das fachliche **WAS** muß präzise spezifiziert sein
- **WIE** etwas implementiert wird, muß nicht spezifiziert werden.
- Präzision muß so gut sein, daß **zwei** Softwareanbieter aus **einer** Spezifikation Software bauen können, die sich identisch verhält und austauschbar ist.

## Nicht ausreichend sind also....

- Nennung von Funktionen
  - Es muß genau beschrieben werden, was die Funktionen tun
  - Die Ein-/ und Ausgabeparameter müssen genau beschrieben werden
- Nennung von Entitäten und Attributen
  - Beziehungen müssen beschrieben werden
  - Attribute müssen typgenau beschrieben sein
  - fachliche Beschreibungen müssen vorhanden sein
- Aufzählung von Anforderungen
  - sind wichtig für den späteren Abgleich mit der spezifizierten Funktionalität

## Ihre Leser sind ...

- Zuerst **Sie** selbst  
    Sie strukturieren damit Ihre Arbeit
- **Entwickler** aus anderen Teilteams
  - die werden Sie reviewen und sich informieren
- **und auch** Mitarbeiter aus der Fachabteilung
  - wir erinnern uns – dass Fachmitarbeiter Datenmodelle lesen und Datentypen definieren ist Wunschdenken



# Vor der Spezifikation des Anwendungskerns sollten Sie ...



- Anforderungen gelistet haben
- Geschäftsprozesse und deren Teilprozesse genannt und beschrieben haben
- Geschäftsvorfälle genannt haben
- Wesentlichen Systemfunktionen katalogisiert haben

# Und wie bitte beschreibe ich einen Anweungskern?



wait and see ✍️

# Spezifikation mit der Sachbearbeiter-Methode



- Sachbearbeiter erbringen Dienste
- Sachbearbeiter haben Gedächtnis
  - Karteikästen
- Sachbearbeiter beauftragen sich gegenseitig
  - Sie als Spezifikateure sind die Organisatoren
- Die Dienste müssen genau spezifiziert werden
  - Aufträge der Sachbearbeiter

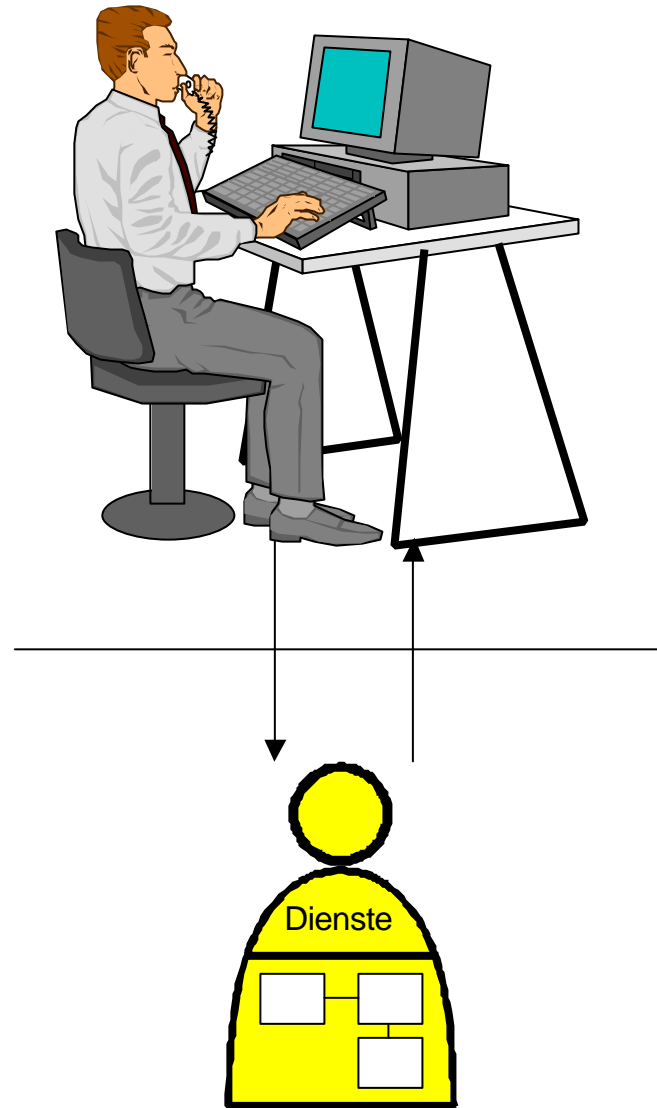
# Spezifikation mit der Sachbearbeiter-Methode



- Die Dinge, die ein Sachbearbeiter verwaltet haben ein Zustandsmodell
- Zusammenhang Zustandsmodell und Dienste
- Der Karteikasten ist eine Datensicht
- Diskussion
  - Verwendbarkeit der Metapher
  - Methodische Einordnung

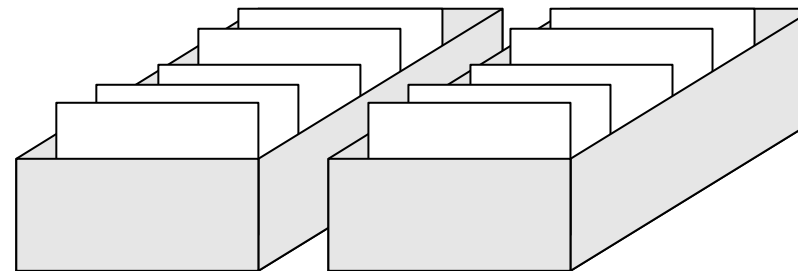
# Sachbearbeiter erbringen Dienste

- EDV-Sachbearbeiter erledigen Aufgaben für menschliche Sachbearbeiter
- Sie erbringen Dienste
- Sie werden aus den Geschäftsprozessen, aus der Benutzungsschnittstelle, aus Batches beauftragt



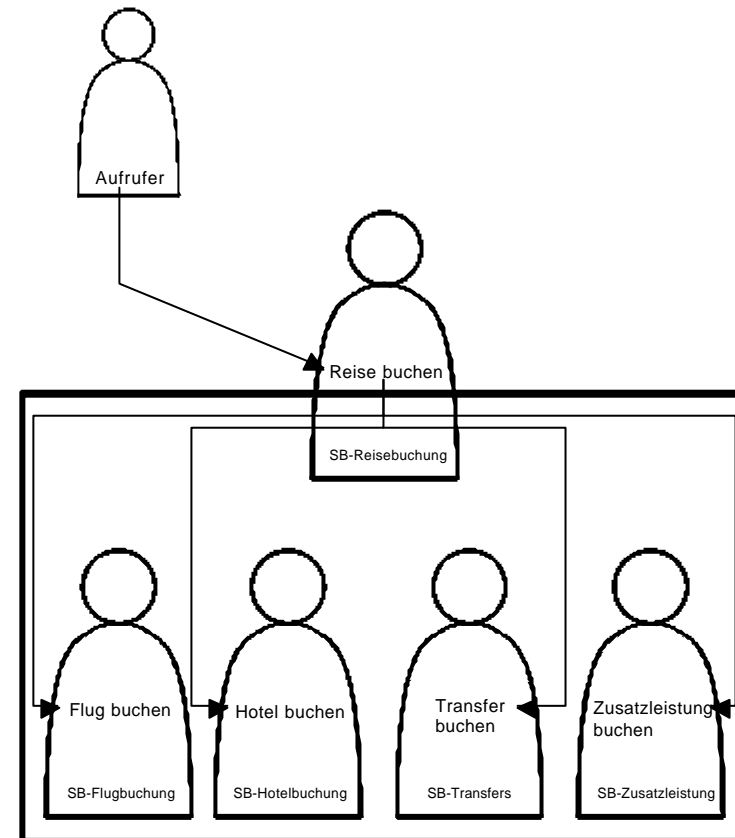
## Sachbearbeiter haben Gedächtnis

- Karteikästen bilden das Gedächtnis der Sachbearbeiter
- Jeder Sachbearbeiter darf nur in seinen Karteikästen schreiben
- Lesen darf er überall  
(mit Einschränkungen)



## Sachbearbeiter beauftragen sich gegenseitig

- Sachbearbeiter können sich Aufträge erteilen
- Aufträge, die direkt aus Geschäftsprozessen kommen, heißen initiale Aufträge oder Geschäftsvorfälle



# Die Dienste müssen genau spezifiziert werden



- Name des Dienstes
- Parameter
- Vor- und Nachbedingungen
- Effektbeschreibung

## Auftrag ablehnen

Braucht	Einstellungsdatum	:	DATE
Braucht	Forderungsnummer	:	Forderungsnummer
Braucht	Begründung	:	Kommentar
Braucht	AutomatischerBrief	:	BOOL

## Vorbedingung

Eine entsprechende Forderung ist existent.

## Vorbedingung

Es handelt sich auch wirklich um einen Anspruch

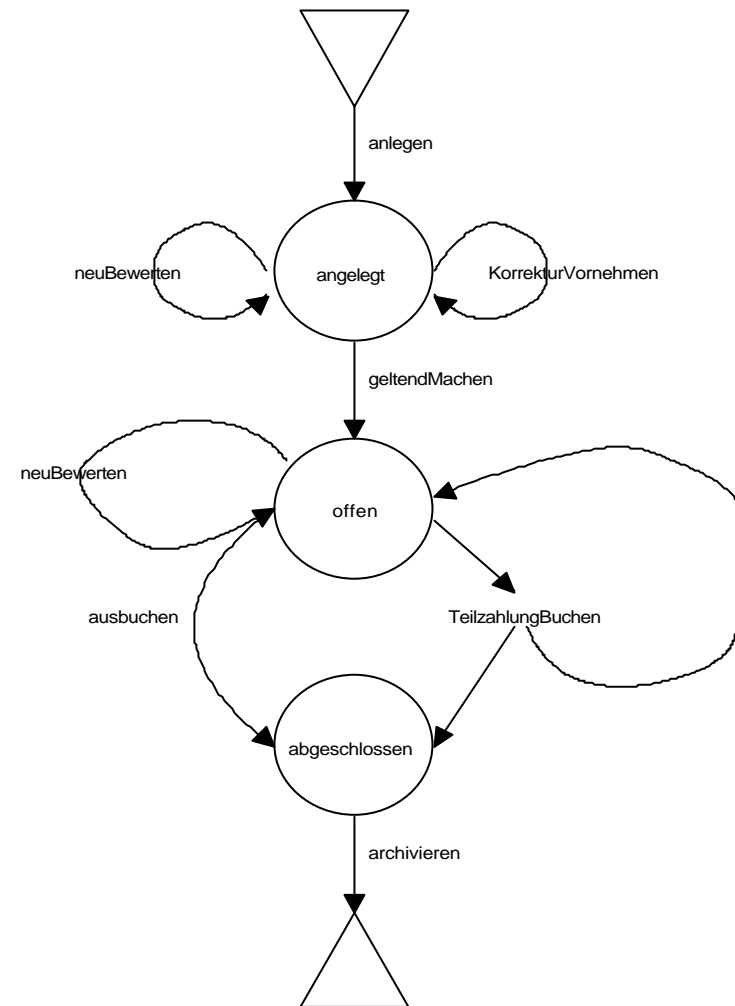
## Effekt

Es wird der Zustand abgelehnt eingenommen. Es wird durch das e Korrespondenzsystem (Sachbearbeiter Korrespondenz.BriefErstellen....) eine der Ablehnung erstellt, falls dies vom Sachbearbeiter gewünscht ist.

Die Begründung wird zusammen mit anderen organisatorischen Da *Forderung.Historie* festgehalten

# Die Dinge, die ein Sachbearbeiter verwaltet, haben ein Zustandsmodell

- Sachbearbeiter verwalten meist Dinge (zum Beispiel Regresse zu Schäden)
- Solche Dinge haben ein Zustandsmodell



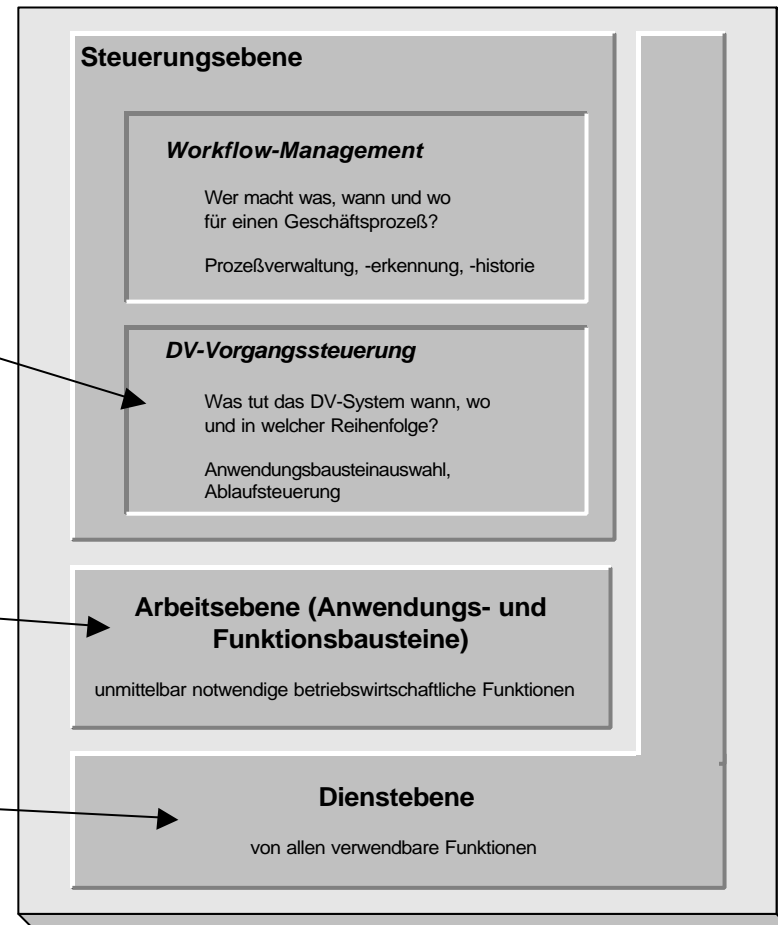
## Zusammenhang Zustandsmodell und Dienste

- Zustandsübergänge werden durch Aufträge bewirkt
- Man findet die Dienste oft über das Nachdenken über ein Zustandsmodell
- Es gibt auch mehr als einen Folgezustand  
Beispiel: Teilzahlung buchen



# Sachbearbeiter und zum Beispiel VAA-Architektur

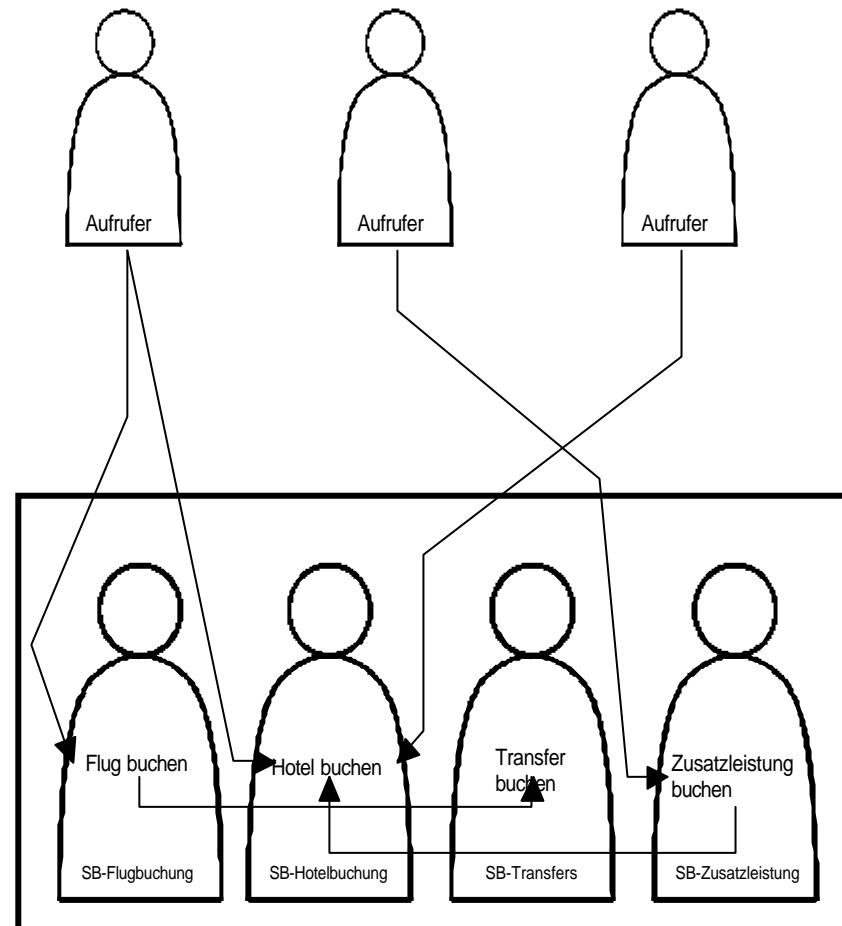
- Die Aufträge kommen aus der DV-Vorgangssteuerung
- Sachbearbeiter bilden die Arbeitsebene der VAA-Architektur
- Das Datenmodell wird im Datenmanager verwaltet. (Dienstebene)



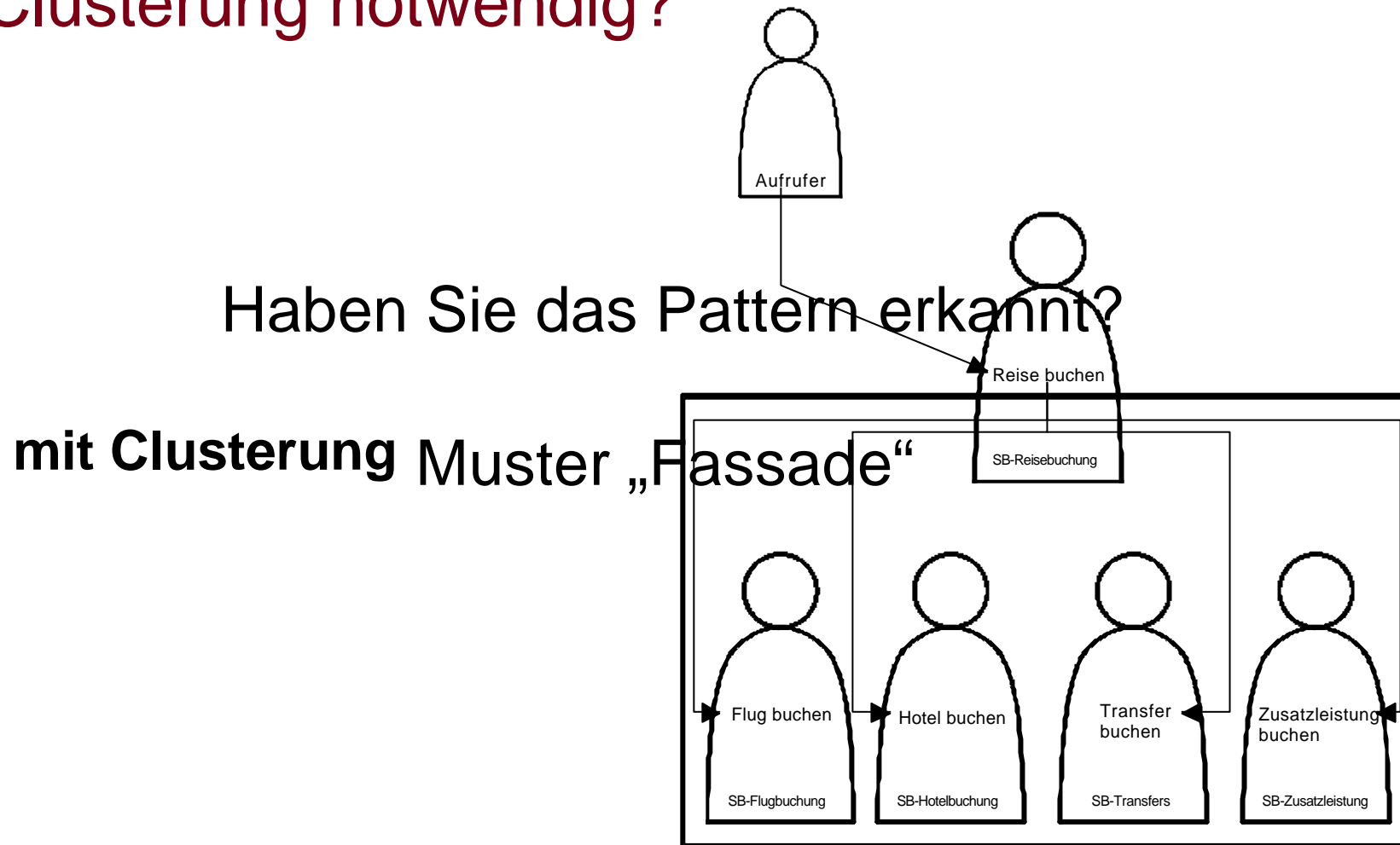
# Sachbearbeiter und austeine: Warum ist Clusterung notwendig?

ohne Clusterung

Änderungen im Team  
brechen Organisation  
der Aufrufer

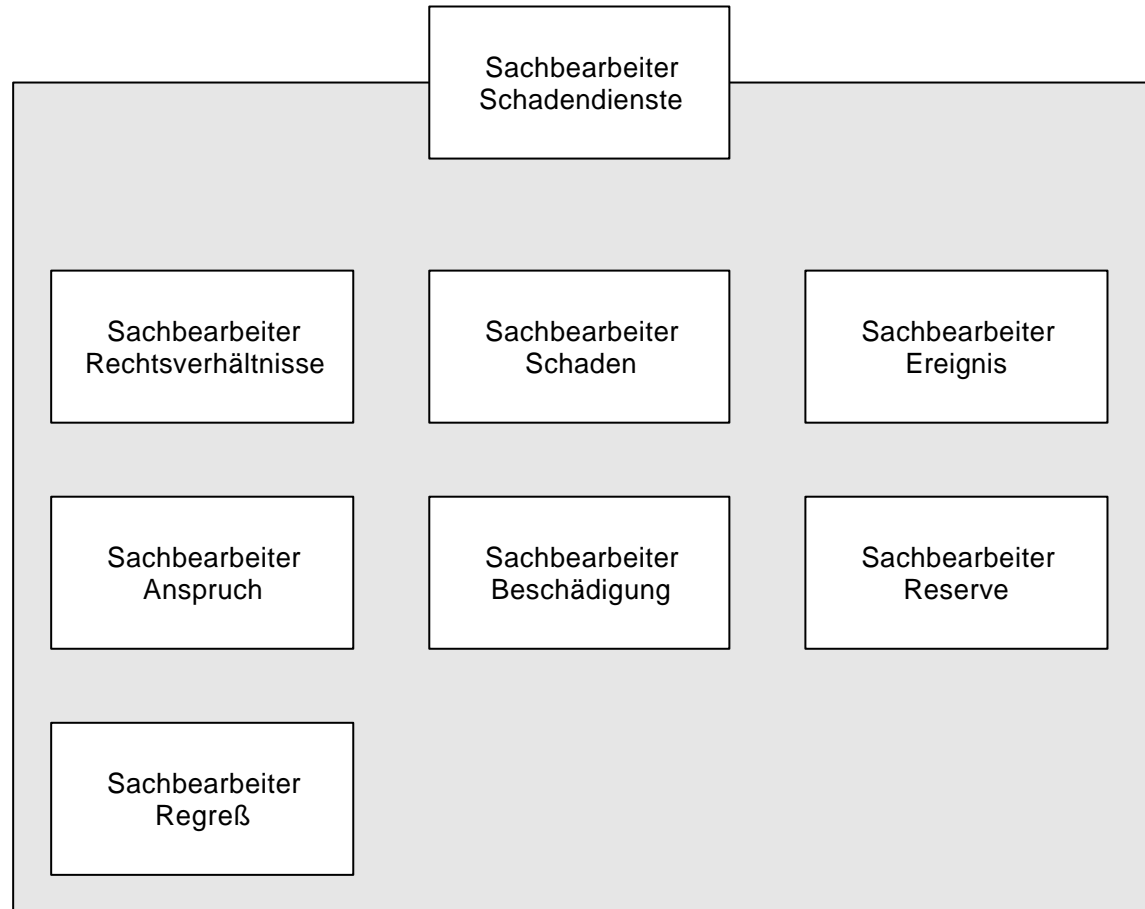


# Sachbearbeiter und VAA-Bausteine: Warum ist Clustering notwendig?

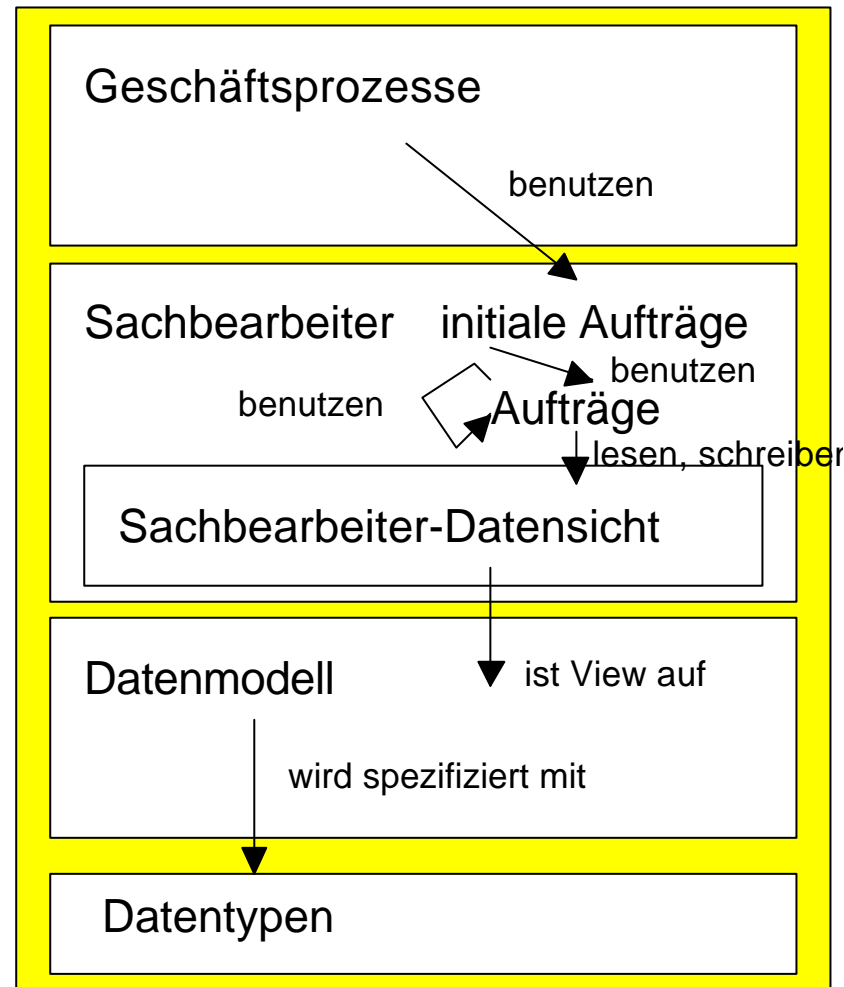


# Sachbearbeiter und Anwendungsbausteine

## Beispiel Schaden/Leistung



# Sachbearbeiter: Zusammenfassung



# Diskussion

## Verwendbarkeit der Metapher



- Sachbearbeiter sind für die Spezifikation technischer Subsysteme nicht geeignet, Grund: Bruch der Metapher
- Sachbearbeiter eignen sich gut für fachliche Bausteine: Sachbearbeiter-Metapher fügt sich gut in "Sachbearbeitung" mit Workflows ein.
- Einordnung in eine Schichten oder J2EE-Architektur einfach
- Verträglich mit Datenzugriffsschichten
- konventionelle Datenmodellierung kann beibehalten werden
- Kapselung durch Sachbearbeiter erhöht Übersicht der Modelle

# Diskussion

## Methodische Einordnung



Methode	SA/SD	Sachbearbeiter	Objektorientiert
Charakteristik	-Daten -Funktionen	+ Kapselung	+ Vererbung + Polymorphie
Einordnung		Soft-OO	Hard-OO
Verwendung		fachliche Spezifikationen (Versicherungs- logik)	Architektur- Spezifikationen (Technik)

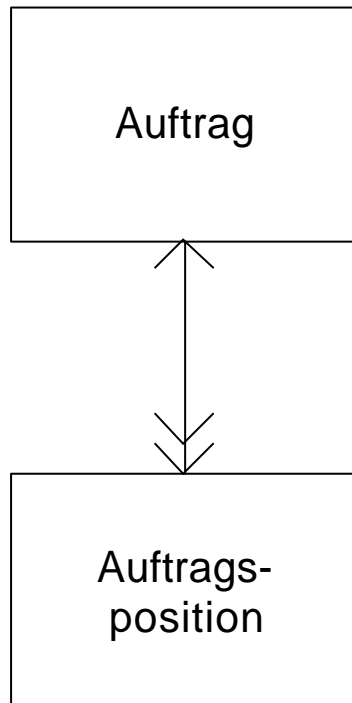
# Beschreibung von Datenmodellen

# Datenmodellierung und die Sachbearbeiter-Methode



- Datenmodellierung soll hier nicht grundlegend eingeführt werden
  - siehe Literatur
  - siehe diverse Datenmodellierungshandbücher
- Übersicht über Konstrukte
- Was man beachten sollte
- Datentypen

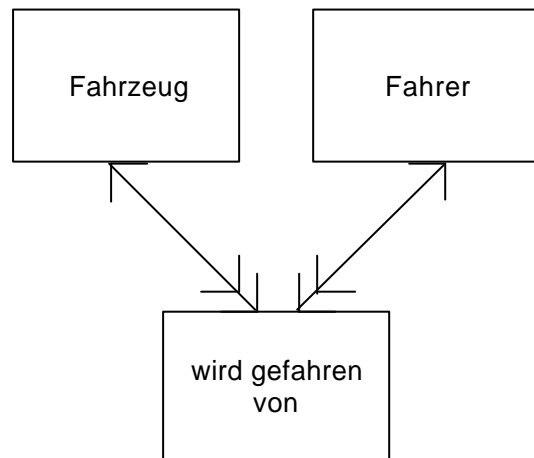
# Übersicht über Konstrukte Datenmodell Graphische Notation



1:n Beziehungen  
einschließlich  
1:0,n; 1:1 etc.

# Übersicht über Konstrukte Datenmodell

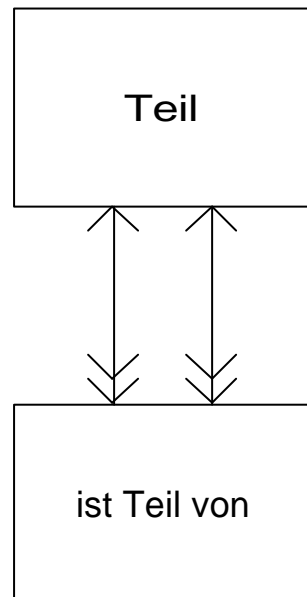
## Graphische Notation



n:m Beziehungen  
einschließlich aller Varianten

# Übersicht über Konstrukte Datenmodell

## Graphische Notation

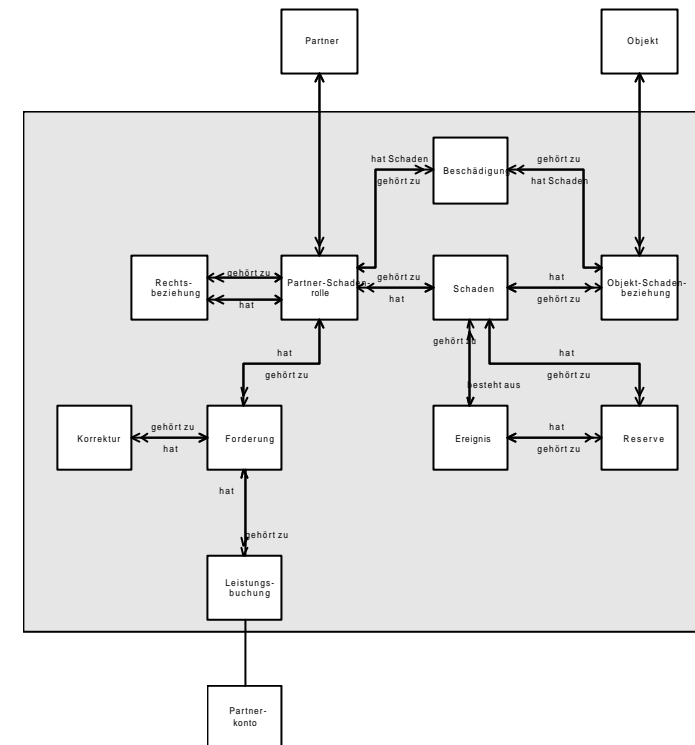


Selbstbeziehung  
(is part of)

# Datenmodellierung

## Was man beachten sollte

- Übersichten müssen textuell erläutert werden
  - jede Entität
  - jede Beziehung
  - Beispiele sind nicht schädlich, sondern hilfreich
- siehe "Beispielspezifikation zum Methodenhandbuch"



# Datenmodellierung

## Elemente der textuellen Beschreibung



(Regel) *Entitätsbeschreibung ::=*

**Entität** *Identifikator*

**Beschreibung**

*Freitext*

**Primärschlüssel**

*Attributname\**

**{Schlüsselkandidat**

*Attributname\* }* \*

**Attribute**

( *Attributdeklaration* | *Beziehungsdeklaration* ) +

(Regel) *Attributdeklaration ::=*

**hat** *AttributName*                      *Arität*      *DatentypName*

*Beschreibungstext*

# Datenmodellierung

## Was man beachten sollte



- Entitäten sollen ausführlich fachlich beschrieben werden.

### Entität Schaden

#### Beschreibung

Zentrales Objekt des Bausteins Schaden/Leistung ist der **Schaden**. Ein Schaden hat dabei selbst nicht weiter viele Eigenschaften oder Attribute. Man kann sich den Schaden eher als einen Anker vorstellen oder einen elektronischen Aktendeckel, unter dem man alle relevanten Informationen zu einem Schadensfall finden kann.

Insbesondere wird der Schaden hier nicht als elektronische Akte modelliert. Was der Sachbearbeiter also in seiner Schadensakte generell an Unterlagen hat, wird hier nicht weiter dargestellt.

Ein Schaden ist immer genau einer Sparte zugeordnet, die sich aus dem Vertrag oder dem Risiko ergibt. Ein Attribut Sparte wäre allerdings redundant, da es aus dem Risiko oder dem Vertrag entnommen werden kann. Wenn ein Schadensereignis mehr als eine Sparte betrifft, so sind entsprechend viele Schadenobjekte anzulegen. Beispiel hierfür ist ein Auffahrunfall eine Vollkasko versicherten Fahrzeuges. Der geschädigte Unfallgegner, dem der Versicherungsnehmer aufgefahren ist, wird als K-Haftpflichtschaden geführt. Der Schaden am Fahrzeug des Versicherungsnehmers wird als K-Kaskoschaden geführt. Es werden zwei Schäden angelegt, die mittels eines Ereignisses gruppiert werden können, damit Gutachter zum Beispiel nur einmal bestellt werden. ....

# Attribute werden durch einen Datentyp und eine Definition beschrieben



nicht so...

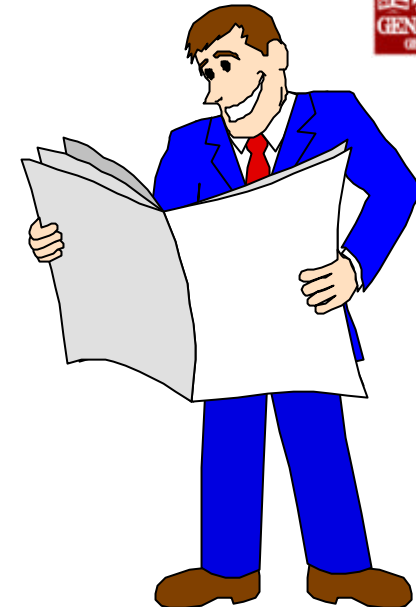
hat Schadendatum

eher so...

hat Schadendatum : DATE

Datum an dem der Schaden eingetreten ist. Dieses Datum ist relevant für die Prüfung der Deckung. Wenn am Schadendatum keine Deckung durch ein Risiko bestand, wenn der Vertrag also aus irgendwelchen Gründen am Schadendatum nicht deckend war, so kann eine Deckung abgelehnt werden.

Das Schadendatum ist im allgemeinen mit dem Datum der Schadensmeldung nicht identisch.



# Datentypkonzept



- Wozu ein Datentypkonzept?
- Elemente
  - Basistypen
  - davon abgeleitete Typen

# Warum ein Datentypkonzept? üblich in der Datenmodellierung

## Attribute

**hat** Kontonummer : PIC 9(5)  
Ordnungsmerkmal für das Konto - künstlicher Schlüssel.

**besser**

## Attribute

**hat** Kontonummer : KtoNummer  
Ordnungsmerkmal für das Konto - künstlicher Schlüssel.

Und zusätzlich den Datentyp KtoNummer

## Datentypen

**DIE** KtoNummer  
**IST EIN** NAT(5)

# Basistypen

Datentyp	Bedeutung
<b>BOOL</b>	Wahrheitswerte, boolesche Werte mit zweiwertiger Logik.  Konstanten <b>TRUE</b> und <b>FALSE</b>
<b>INT(&lt;n&gt;)</b>	Ganzzahl der Länge <n> mit Vorzeichen, zum Beispiel <b>INT(5)</b>
<b>NAT(&lt;n&gt;)</b>	Natürliche Zahl der Länge <n> ohne Vorzeichen, zum Beispiel <b>NAT(3)</b>
<b>FIX(&lt;n&gt;,&lt;m&gt;)</b>	BCD-Arithmetik mit <n> Vorkommastellen und <m> Nachkommastellen.
<b>FLOAT(&lt;n&gt;)</b>	Gleitkommazahl mit n-stelliger Genauigkeit.
<b>STRING(&lt;n&gt;)</b>	Zeichenkette mit <n> Zeichen Länge. Zum Beispiel <b>STRING(10)</b> für eine Zeichenkette der Länge 10.  Konstante <b>EMPTY</b> für die leere Zeichenkette.
<b>DATE</b>	Datum der Form TT/MM/JJJJ
<b>TIME</b>	Zeit in der Form HH:MM:SS

# Abgeleitete Typen



- Aufzählungstypen
- Feldtypen
- Einschränkungstypen
- Verbundtypen

**Beispiele siehe Handbuch und Musterspezifikation**

# Modellierung von Geschäftsprozessen

# Beschreibung von Geschäftsprozessen

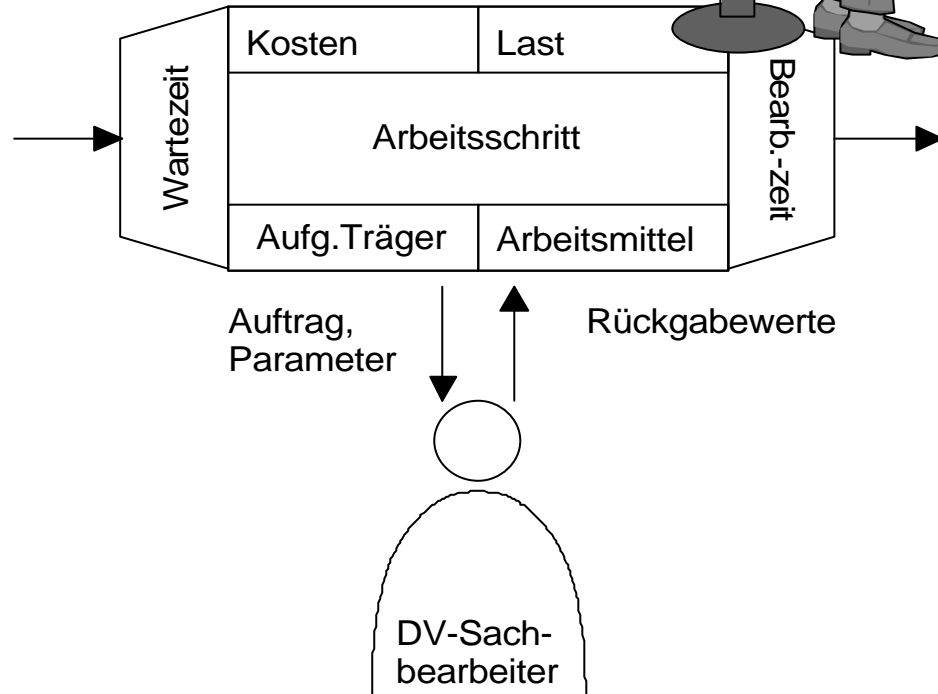
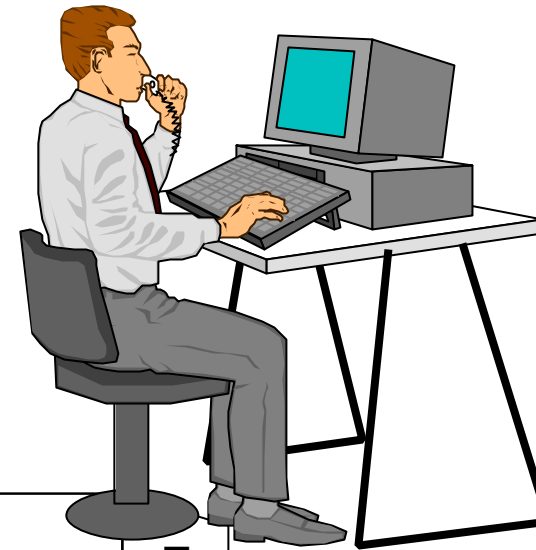
- Sinn und Unsinn der Geschäftsprozeßmodellierung
  - Wann macht GP-Modellierung Sinn?
  - Was ist ein Arbeitsschritt? Was ist ein Prozeß/Teilprozeß?
  - Beispiel
- Die graphischen Beschreibungselemente
- Beschreibung als Text
- Relevanz für Schaden/Leistung

# Ziele GP-Modellierung



- Allgemein: Denken in Vorgängen
  - über Abteilungsgrenzen hinweg.
  - über Grenzen von Anwendungssystemen hinweg.
  - Unterstützung Reengineering.
  - Kostenrechnung, Organisation, weiteres.....

# Zusammenhang: Geschäftsprozesse EDV-Sachbearbeiter



# Sinn und Unsinn

## Geschäftsprozeßmodell NEIN für ...



- Prozeß "Sicherung des Geschäftserfolges"
- Prozeß "Strategische Informationssystemplanung"
  
- Kriterien:
  - keine Unterstützung durch EDV
  - nicht strukturierbar
  - kaum maschinell unterstützbar

# Sinn und Unsinn

## Geschäftsprozeßmodell JA für ...



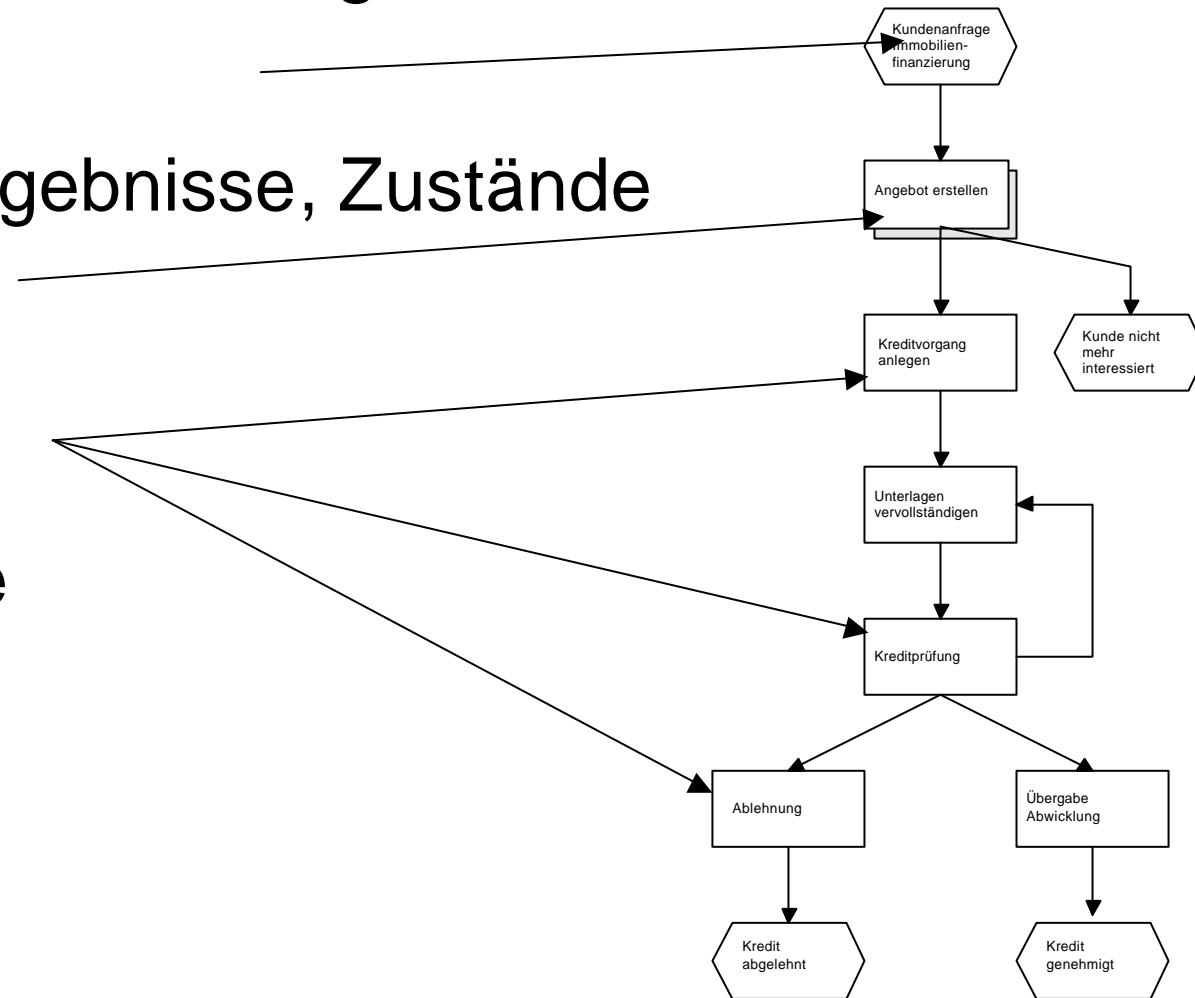
- Mehrstufige Bearbeitung von Immobilienkrediten
- Kriterien:
  - Man kann sich eine Akte vorstellen, die durch die Organisation wandert
  - Die Akte besucht verschiedene menschliche Sachbearbeiter
  - Die Bearbeitung ist strukturierbar
  - Die Bearbeitung wird teilweise durch EDV unterstützt

# Abgrenzung

- Wann ist es ein
  - Geschäftsprozeß?
  - Teilprozeß?
  - Arbeitsschritt?
- Kriterien:
  - Geschäftsprozesse werden in keinem anderen Prozeß als "Teilprozeß" verwendet
  - Arbeitsschritte
    - Wechsel des Bearbeiters
    - Wechsel des Bearbeitungsmittels
    - Wechsel der Phase: Planung, Durchführung, Kontrolle
  - siehe betriebswirtschaftliche Organisationslehre

# Graphische Beschreibungselemente

- Ereignisse, Ergebnisse, Zustände
- Teilprozesse
- Arbeitsschritte



## **Arbeitsschritt** Kreditvorgang anlegen

### **Auslöser**

*Bemerkung: Auslöser ist hier der Abschluß des vorhergehenden Arbeitsschrittes - ein solch „trivialer Auslöser“ sollte besser gar nicht erst beschrieben werden.*

**Bearbeiter** Kreditsachbearbeiter

**Arbeitsmittel** EDV-System Kreditbearbeitung

### **Beschreibung**

Ein Angebot ist erstellt - der Kunde möchte das Angebot annehmen. Die Eckdaten des Angebotes werden im Kreditbearbeitungssystem erfaßt (**Sachbearbeiterauftrag** Kreditvorgang.erfassen).

## **Arbeitsschritt** Unterlagen vervollständigen

**Bearbeiter** Kreditsachbearbeiter

**Arbeitsmittel** Scanner, EDV-System Kreditbearbeitung

### **Beschreibung**

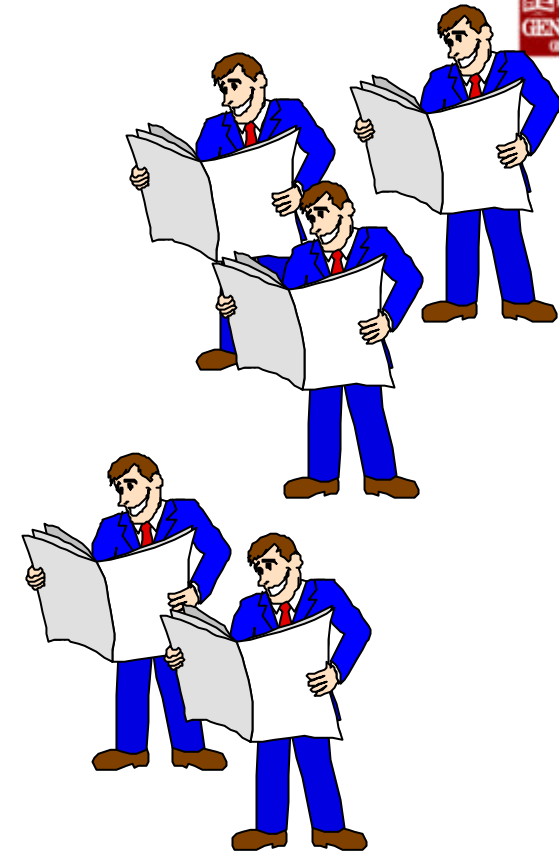
Abhängig von den Eckdaten ermittelt das Kreditbearbeitungssystem die benötigten Unterlagen, die von Wertgrenzen abhängen. (**Sachbearbeiterauftrag** Kreditvorgang.UnterlagenErmitteln).

Unterlagen werden vom Kunden eingereicht und mit dem **Sachbearbeiterauftrag** Kreditvorgang.UnterlageZuordnen dem Vorgang zugeschrieben. Sind alle Unterlagen vorhanden, so erfolgt eine Freigabe mit dem **Sachbearbeiterauftrag** Kreditvorgang.UnterlageZurPrüfungWeiterreichen.

# Eine Mustergliederung für Spezifikationen fachlicher Anwendungsbausteine

## Mustergliederung - Grundidee

- Fachliche Beschreibung
  - Anforderungen
  - Lösungsalternativen
  - Beschreibung des Fachgebietes
  - Szenarien für die Benutzung
- "formale" Beschreibung
  - Geschäftsprozesse
  - Sachbearbeitermodell
  - Datenmodell
  - Datentypen



unterschiedliche Leser

# Mustergliederung



## Mustergliederung

- Überblick
  - fachlicher Teil
  - "formaler Teil"

## Verbindlichkeit der Mustergliederung

- Konformität zu einer Mustergliederung macht noch keine gute Spezifikation
  - Qualitätssicherung ist wichtiger
  - Formalismus ersetzt kein Nachdenken über die Inhalte
  - Es gibt "formal richtige" Spezifikationen, die eine miserable Qualität haben
- Die Mustergliederung ist ein Leitfaden und eine (unvollständige) Checkliste
- Die Mustergliederung muß nicht immer 1:1 eingehalten werden
- Gute Beispiele sind wichtiger

# Mustergliederung

## Überblick "Fachlicher Teil"



- 1 Überblick und Abgrenzung des Anwendungsbausteins
  - 1.1 Fachlicher Überblick
  - 1.2 Abgrenzung
- 2 Fachliche Beschreibung des Anwendungsbausteins
  - 2.1 Wesentliche Anforderungen
  - 2.2 Lösungsalternativen
  - 2.3 Beschreibung des Bausteins
- 3 Szenarien

# Mustergliederung

## Überblick "Formaler Teil"



- 4 Formale Beschreibung
  - 4.1 Geschäftsprozesse
    - 4.1.1 Geschäftsprozeß <Beispiel 1>
    - 4.1.n Geschäftsprozeß <Beispiel n>
  - 4.2 Sachbearbeitermodell
    - 4.2.1 Sachbearbeiter <Beispiel 1>
    - 4.2.n Sachbearbeiter <Beispiel n>
  - 4.3 Batches
    - 4.3.1 Batch <Batch 1>
    - 4.3.n Batch <Batch n>

# Mustergliederung

## Überblick "Formaler Teil" (Forts.)



- 4.4            Datenmodell
- 4.4.1        Überblick
- 4.4.2        Entität <Entität 1>
- 4.4.n        Entität <Entität n>
  
- 4.5            Datentypen
  
- 4.6            Offene Schnittstellen

# Die Sachbearbeitermethode und OO

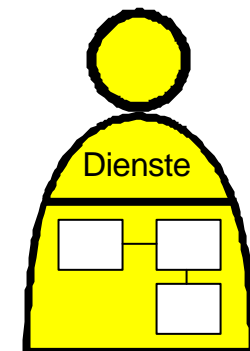
In der Sachbearbeiter-Methode stecken viele Patterns – ohne dass das 1990 bewußt war

## Die Sachbearbeitermethode und OO

- Hartes OO enthält zusätzlich noch Vererbung und Polymorphie ..
- Implementierungsvererbung in „fachlichen Systemen“ wurde inzwischen als „schädlich“ beschrieben – siehe auch J2EE Patterns
- Polymorphie, sofern sie „fachlich“ ist, fällt ebenfalls darunter

# Objektorientierte Methoden: Beispiel UML

- UML: Grafikorientiert
  - Anwendungsfälle (Use Cases)
  - Sequenzdiagramm:  
Modellierung objektübergreifender Abläufe
  - plus Beschreibung der Objekttypen
    - und pro Objekt Zustandsdiagramm:  
Modellierung objektinterner Abläufe
  - Typenmodell



# Sachbearbeiter sind nicht von gestern ✍️

## Man findet darin folgende J2EE Patterns



- Man kann den Sachbearbeiter auch als Business Delegate bezeichnen
  - <http://developer.java.sun.com/developer/restricted/patterns/BusinessDelegate.html>
- Die Fassade, die wir gesehen haben, heißt dort Session Facade
  - <http://developer.java.sun.com/developer/restricted/patterns/SessionFacade.html>
- Aggregate Entity entspricht der Datensicht eines Sachbearbeiters
  - <http://developer.java.sun.com/developer/restricted/patterns/AggregateEntity.html>
- Die Aufträge kann man mit Value Objects implementieren
  - <http://developer.java.sun.com/developer/restricted/patterns/ValueObject.html>